

Claims

1. A method of executing program code in a secure manner in a data processor, comprising:
 - 5 fetching an instruction for execution from a memory;
 - determining that the instruction has access privileges for accessing a specified location within the memory; and
 - accessing the specified location only when the instruction has privileges for accessing the specified location.
- 10 2. The method of claim 1 wherein determining comprises comparing a privilege for the instruction to a level of privilege required to access the specified location.
- 15 3. The method of claim 1 wherein determining comprises comparing a privilege for the instruction to a level of privilege required to access the specified location by control unit, wherein the control unit performs acts of:
 - accepting a virtual address from the instruction;
 - accepting a first privilege level from the instruction;
 - 20 converting the virtual address to a physical address corresponding to the specified location;

looking up a second privilege level required in order to access the specified location;

comparing the second privilege level to the first privilege level; and

granting access to the instruction only when the first privilege levels meets

5 or exceeds a threshold privilege level determined by the second privilege level.

4. The method of claim 1 wherein determining comprises comparing a privilege for the instruction to a level of privilege required to access the specified location by control unit, wherein the control unit performs acts of:

10 accepting a virtual address from the instruction;

accepting a first privilege level from the instruction;

converting the virtual address to a physical address corresponding to the specified location;

15 looking up a second privilege level required in order to access the specified location;

comparing the second privilege level to the first privilege level;

granting access to the instruction only when the first privilege levels meets or exceeds a threshold privilege level determined by the second privilege level; and

20 halting execution of the instruction when the first privilege level does not meet or exceed a threshold privilege level determined by the second privilege level.

5. The method of claim 1, wherein the specified location is a secure region of the memory.

5 6. The method of claim 1 where the secure region comprises a range of addresses of the memory.

7. A method of executing program code in a secure manner in a data processor, comprising:

10 fetching an instruction for execution;

determining that the instruction that the instruction has access privileges for accessing a specified location within a memory; and

accessing the specified location only when the instruction has privileges for accessing the specified location, where the determining is performed in part by

15 converting the specified location into a physical address via a memory control unit.

8. The method of claim 7 further comprising disabling interrupts before fetching the instruction.

20

9. The method of claim 7 wherein the memory control unit controls all access to the memory by any instruction.

10. The method of claim 7 where the accessing the specified location comprises accessing code in a secure portion of the memory.

5 11. The method of claim 7 further comprising:

comparing the specified location with a set of predetermined entry locations;

executing the instruction at the specified location only if it is contained in the set of locations, wherein the set of locations corresponds to a table of physical
10 addresses and corresponding access privileges.

12. A method of executing program code in a secure manner in a data processor, comprising:

fetching an instruction for execution;

15 determining that the instruction accesses a specified location within a secure region of the memory;

accessing the specified location only when the instruction is accompanied by corresponding current privilege level data, where the determining is carried at least in part via conversion of the specified location to a physical address in the
20 memory; and further comprising:

comparing the specified location with a set of predetermined entry locations;

executing the instruction at the second location only if it is contained in the set of locations;

comparing the current privilege level with a predetermined required privilege level associated with the second location;

5 executing the instruction at the second location only if the current privilege level is at least as high as the required privilege level.

13. A method of executing program code in a secure manner in a data processor, comprising:

10 fetching a sequence of instructions in the code, the sequence of instructions including a privilege level associated with the sequence;

 determining virtual addresses that the code accesses;

 converting, by a control logic unit, the specific addresses to corresponding physical addresses;

15 accessing the secure memory region only when privilege level associated with the sequence equals or exceeds a privilege level associated with the physical addresses; and

 executing at least a part of the sequence atomically.

20 14. The method of claim 13 where executing at least part of the sequence atomically comprises replacing a normal interrupt handler with another

handler that prevents accesses to the physical addresses during execution of the code.

15. The method of claim 13 where executing at least part of the
5 sequence atomically comprises restricting the operation of processor interrupts to a processor executing the code while the sequence of instructions is executing.

16. The method of claim 13 where executing at least part of the
sequence atomically comprises preventing processor interrupts to a processor
10 executing the code while the sequence of instructions is executing.

17. A method of executing program code in a secure manner in a data processor, comprising:

fetching a sequence of instructions in the code, the sequence of instructions
15 including a privilege level associated with the sequence;

determining virtual addresses that the code accesses;

converting, by a control logic unit, the specific addresses to corresponding
physical addresses;

determining that the physical addresses correspond to a secure region of a
20 memory;

accessing the secure memory region only when privilege level associated with the sequence equals or exceeds a privilege level associated with the physical addresses; and

destroying at least some data upon occurrence of a specified event.

5

18. The method of claim 17 wherein the destroyed data comprises contents of at least some locations in the secure memory.

19. The method of claim 17 wherein the destroyed data comprises
10 contents of at least one register of a processor executing the code.

20. The method of claim 17 where the event is an interrupt sent to a processor executing the code.

15 21. The method of claim 17 where the event is a reboot of the processor executing the code.

22. The method of claim 17 where the event is an attempt by a device external to the processor executing the code to access the secure memory region.

20

23. A method of executing program code in a secure manner in a data processor, comprising:

fetching a sequence of instructions in the code, the sequence of instructions including a privilege level associated with the sequence;

5 determining virtual addresses that the code accesses;

converting, by a control logic unit, the specific addresses to corresponding physical addresses;

determining that the physical addresses correspond to a secure region of a memory;

10 accessing the secure memory region only when privilege level associated with the sequence equals or exceeds a privilege level associated with the physical addresses; and

restricting access to the secure memory region by devices external to a processor executing the code.

15

24. The method of claim 23 where access is restricted during execution of the code.

25. The method of claim 23 where restricting access to the secure
20 memory region comprises locking a memory bus coupled to the memory.

26. The method of claim 23 where restricting access to the secure memory region comprises preventing a bus master from accessing the region.

27. A method of executing program code in a secure manner in a data processor, comprising:

5 fetching a sequence of instructions in the code;

 determining specific addresses that the code accesses;

 converting, by a control logic unit, the specific addresses to corresponding physical addresses;

10 determining privilege levels required in order to access the respective physical addresses;

 comparing the determined privilege levels to privilege levels associated with the sequence of instructions; and

 accessing the secure memory region only when the determined privilege

15 levels meet or exceed a threshold privilege level determined from the associated privilege levels.

28. A method of executing program code in a secure manner in a data processor, comprising:

20 fetching code comprising a sequence of instructions , the sequence of instructions including a privilege level associated with the sequence;

 determining virtual addresses that the code accesses;

converting, by a control logic unit, the specific addresses to corresponding physical addresses;

determining that the physical addresses correspond to one of multiple secure rings within the memory;

5 accessing the first ring only if the sequence includes a privilege level corresponding to the first ring to a ring higher in an hierarchy of the multiple secure rings of the memory.

29. The method of claim 28 where the secure memory region comprises
10 a range of addresses in the memory.

30. The method of claim 28 where the secure rings comprise ranges of addresses within an address range of the secure memory region.

15 31. The method of claim 28 where the hierarchy has two secure levels within an outer unsecure level.

32. The method of claim 31 where one of the secure rings is higher in the hierarchy than the other ring.

20

33. The method of claim 28 where the memory has at least first and second subrings within one of the secure rings, and further comprising:

determining whether the code accesses the first subring within the first ring;

accessing the first subring only if the code is located within the first subring

5 of the one ring;

determining whether the code accesses the second subring of the one ring;

and

accessing the second subring only if the code is located within the second

subring of the one ring.

10

34. The method of claim 33 further comprising:

determining whether the code accesses the one ring outside both the first

and the second subrings; and

accessing the one ring outside both the first and the second subrings of the

15 first ring if the code is located within either the first or the second subring of the one ring.

35. The method of claim 32 where another of the secure rings is inner to the one ring, and further comprising:

determining whether the code accesses the one ring, including the first and second subrings thereof; and

5 accessing the one ring, including the first and second subrings, if the code is located in the other, inner ring.

36. A medium carrying computer readable representations for causing a computer to carry out the method of claim 28.

10

37. A data processor for executing secure code residing in a memory, comprising:

an instruction decoder for determining that a current instruction has an associated privilege level appropriate to a secure portion of a memory;

15 an instruction pointer for holding an address of a current instruction in the memory; and

control logic coupled to the instruction decoder for executing the current instruction only when the associated privilege level corresponds to one or more predetermined regions of the memory.

20

38. The data processor of claim 37 where at least one of the predetermined memory regions is defined by a range of addresses in the memory.

39. A data processor for executing secure code residing in a memory,
comprising:

- an instruction decoder for determining that a current instruction has an
- 5 associated privilege level appropriate to a secure portion of a memory;
- an instruction pointer for holding an address of a current instruction in the
- memory; and
- control logic coupled to the instruction decoder for executing the current
- instruction only when the associated privilege level is appropriate to the secure
- 10 portion of a memory, where at least a portion of one of the predetermined memory
- regions is implemented in a technology different from that of the remainder of the
- same portion.

40. A data processor for executing secure code residing in a memory,
15 comprising:

- an instruction decoder for determining that a current instruction has an
- associated privilege level appropriate to a secure portion of a memory;
- an instruction pointer for holding an address of a current instruction in the
- memory;
- 20 control logic coupled to the instruction decoder for executing the current
- instruction only when the associated privilege level is appropriate to the secure
- portion of a memory, where at least a portion of one of the predetermined memory

regions is implemented in a technology different from that of at least a portion of another one of the regions.

41. A data processor for executing secure code residing in a memory,
5 comprising:

an instruction decoder for determining that a current instruction has an associated privilege level appropriate to a secure portion of a memory;

an instruction pointer for holding an address of a current instruction in the memory;

10 control logic coupled to the instruction decoder for executing the current instruction only when the associated privilege level is appropriate to the secure portion of a memory, where the memory is on the same module with the instruction decoder, the instruction pointer, and the control logic.

15 42. The data processor of claim 41 where the memory is on the same integrated-circuit chip with the instruction decoder, the instruction pointer, and the control logic.

43. The data processor of claim 41 where the memory includes a flash
20 memory for holding the secure code.

44. The data processor of claim 43 where the memory further includes read/write memory accessible to the secure code.

45. The data processor of claim 44 where the instruction decoder
5 responds to one of a defined set of distinguished operation codes for identifying the current instruction as accessing secure code.

46. The data processor of claim 45 where the instruction decoder
executes a current instruction having one of the distinguished operation codes only
10 when the current instruction matches one of a set of defined target locations in the memory.

47. A data processor for executing secure code residing in a memory,
comprising:

15 an instruction decoder for determining that a current instruction has an associated privilege level appropriate to a secure portion of a memory;

an instruction pointer for holding an address of a current instruction in the memory;

control logic coupled to the instruction decoder for executing the current
20 instruction only when the associated privilege level is appropriate to the secure portion of the memory, where the instruction decoder responds to one of a defined set of distinguished operation codes for identifying the current instruction as

accessing secure code, where the processor operates at multiple different privilege levels, and where the instruction decoder executes a current instruction having at least one of the distinguished operation codes only if the processor is currently operating at a particular one of the levels.

5

48. A data processor for executing secure code residing in a memory, comprising:

an instruction decoder for determining that a current instruction belongs to the secure code when the current instruction has an associated privilege level

10 appropriate to a secure portion of a memory;

an instruction pointer for holding an address of a current instruction in the memory;

control logic coupled to the instruction decoder for executing the current instruction only when the associated privilege level is appropriate to the secure

15 portion of the memory, and further comprising curtain logic coupled to the instruction decoder for restricting access to a predetermined range of addresses in the memory by any instruction not belonging to the secure code.

49. The data processor of claim 48 further comprising a bus lock
20 responsive to the curtain logic for prohibiting access to the predetermined address range during execution of the secure code.

50. The data processor of claim 49 where the system includes at least one bus master external to the processor, and where the bus lock disables any bus master during execution of the secure code.

5 51. A data processor for executing secure code residing in a memory, comprising:

an instruction decoder for determining that a current instruction belongs to the secure code when the current instruction has an associated privilege level appropriate to a secure portion of a memory;

10 an instruction pointer for holding an address of a current instruction in the memory;

control logic coupled to the instruction decoder for executing the current instruction only when the associated privilege level is appropriate to the secure portion of the memory, and further comprising an interrupt handler for restricting
15 processing of interrupts during execution of the secure code.

52. The data processor of claim 51 where the interrupt handler disables interrupts during execution of the secure code.

20 53. The data processor of claim 51 where the interrupt handler disallows devices external to the processor from accessing at least one of the predetermined memory regions during execution of the secure code.

54. A medium bearing a computer readable representation configured to cause a processor to execute curtained code, wherein the computer readable representation is further configured to cause the processor to execute the curtained
5 code in response to determining that the curtained code corresponds to a privilege level associated with physical addresses corresponding to virtual addresses accessed by the curtained code.

55. The medium of claim 54, wherein the computer readable
10 representation is further configured to cause the processor to execute the curtained code from a curtained portion of a memory having multiple portions each bearing a respective security curtain level.

56. The medium of claim 54, wherein the computer readable
15 representation is further configured to cause the processor to execute the curtained code from a curtained portion of a memory that also includes open portions exclusive of the curtained portion.

57. The medium of claim 54, wherein the computer readable
20 representation is further configured to cause the processor to execute the curtained code from a predetermined portion of a memory comprising multiple segregated

curtained portions each requiring a different access privilege level to be associated with the code accessing the multiple portions.

58. The medium of claim 54, wherein the computer readable
5 representation is further configured to cause the processor to execute the curtained code atomically.

59. The medium of claim 54, wherein the computer readable
representation configured to cause a processor to execute curtained code
10 comprises a computer readable representation configured to:

fetch a sequence of instructions in the code;

determine that the sequence has an associated privilege level appropriate to
a secure portion of a memory;

determine that the code accesses the secure region;

15 access the secure memory region only when the associated privilege level is
appropriate to the secure portion of the memory; and

destroying at least some data upon occurrence of a specified event.

60. The medium of claim 54, wherein the computer readable
20 representation configured to cause a processor to execute curtained code
comprises a computer readable representation configured to:

fetch a sequence of instructions in the code;

determine that the sequence has an associated privilege level appropriate to a secure portion of a memory;

determine that the code accesses the secure region of a memory;

access the secure memory region only when the associated privilege level is
5 appropriate to the secure portion of the memory;

destroy at least some data upon occurrence of an interrupt sent to a processor executing the code.